

Icy Foodies Project Documentation

Introduction

"Icy Foodies" is a comprehensive online food ordering platform designed for the interaction of users with food services. So, what it does is that it allows users to browse, customize, and order food items. The application integrates modern web technologies to ensure a smooth and efficient user experience, catering to the needs of food enthusiasts while also benefiting food businesses through innovative features like loyalty programs and order tracking.

Project Overview

"Icy Foodies" allows users to explore a wide variety of food items, view detailed nutritional information, customize their orders, and manage their shopping cart efficiently. The project is developed as a full-stack application, with React for the frontend, Node.js for the backend, and PostgreSQL for data storage. Deployed on a cloud platform, "Icy Foodies" is accessible via desktop and mobile devices, ensuring convenience and accessibility for all users.

Core Objectives

- To provide a user-friendly platform for ordering food online.
- To incorporate advanced features like order tracking and loyalty rewards.
- To ensure secure and efficient user authentication and data management.

Deployment

The live application is accessible at the following link: <https://frontend-oj2cibmwy-cesonus-projects.vercel.app/>

Features and Functionality

User Authentication

The platform incorporates secure user authentication features, allowing users to:

- Sign up for a new account.
- Log in with existing credentials.
- Update their profiles, including personal preferences.

Test Login Credentials:

- **Username:** Arnold.yhaw@gmail.com
- **Password:** 1234567

Menu Display

Users can browse a variety of food items, each presented with:

- Detailed descriptions.
- Pricing information.
- Customization options (e.g., extra toppings, no onions).
- Nutritional details (e.g., calories, fat content, allergens).

Cart Management

The cart system is designed to provide a smooth checkout experience. Users can:

- Add items to their cart and adjust quantities dynamically.
- Remove items with ease.
- Apply promo codes for discounts.
- View a detailed breakdown of costs, including delivery charges.

Order Placement and Tracking

Users can place their orders with just a few clicks and track their status in real-time:

- Order statuses include "Preparing," "Ready for Pickup," and "Delivered."
- Users can check the status of their orders at any time.

Loyalty Program

The loyalty program rewards users for their purchases:

- Users earn points for each order based on the total price.
- Points can be redeemed for discounts on future orders.
- A user-friendly interface displays available points and allows redemption.

Feature Checklist

Below is a list of all implemented features:

- User Authentication: Login, Signup, and secure access.
- Menu Display: Food items with customization options and nutritional information.

- Add to Cart: Ability to add items to the cart and update quantities.
- Cart Management: Remove items, apply promo codes, and calculate totals.
- Order Placement: Place orders with payment simulation.
- Order Tracking: View order status (e.g., Preparing, Ready for Pickup).
- Loyalty Program: Earn points for orders and redeem them for discounts.
- Responsive Design: Optimized for mobile and desktop devices.

Responsive Design

"Icy Foodies" is fully responsive, ensuring a consistent experience across devices, from desktops to smartphones.

Technology Stack

Frontend

- **React**: For building the user interface and handling dynamic data.
- **CSS**: For styling and ensuring a visually appealing design.

Backend

- **Node.js**: For handling server-side logic and APIs.
- **Express.js**: For creating and managing RESTful endpoints.

Database

- **PostgreSQL**: For reliable and efficient data storage and management.

Installation and Local Setup

To run the project locally, follow these steps:

Prerequisites

- Install Node.js (v16 or later) and PostgreSQL.
- Ensure Git is installed on your machine.

Backend Setup

1. Clone the repository:

```
git clone https://github.com/your-repo/icy-foodies.git  
cd icy-foodies
```

2. Navigate to the `backend` directory and install dependencies:

```
cd backend  
npm install
```

3. Configure the environment variables in a `.env` file:

```
DB_USER=your_database_user  
DB_PASSWORD=your_database_password  
DB_HOST=your_database_host  
DB_DATABASE=icy_foodies  
DB_PORT=5432  
DB_SSL=false
```

4. Initialize the database:

```
npm run db:init
```

5. Start the backend server:

```
npm start
```

The server will run on `http://localhost:4000`.

Frontend Setup

1. Navigate to the `frontend` directory and install dependencies:

```
cd ../frontend  
npm install
```

2. Start the frontend application:

```
npm start
```

The application will run on `http://localhost:3000`.

API Documentation

The backend provides RESTful APIs for various functionalities. Below is an overview of the key endpoints:

User Authentication

- **POST /signup:** Register a new user.

- **POST /login:** Authenticate an existing user.
- **GET /profile/:user_id:** Fetch user profile details.

Menu Items

- **GET /menu-items:** Retrieve all menu items.
- **PUT /menu-item/:food_id:** Update a menu item.
- **DELETE /menu-items/:food_id:** Remove a menu item.

Cart Management

- **POST /cart:** Add an item to the cart.
- **GET /cart/:user_id:** View the cart for a user.

Orders

- **POST /orders:** Place a new order.
- **GET /orders/:order_id/status:** Fetch the status of an order.
- **PUT /orders/:order_id/status:** Update the status of an order.

Loyalty Program

- **GET /users/:user_id/points:** Retrieve loyalty points for a user.
- **POST /users/:user_id/redeem:** Redeem points for discounts.

Screenshots

API Testing

Below is a screenshot of API testing using Postman:

The image displays two separate screenshots of the Postman application interface, both set against a dark theme.

Screenshot 1: Sign-up Request

- Request URL:** https://backend-2-i0ej.onrender.com/signup
- Method:** POST
- Body (raw JSON):**

```
1 {
2   "name": "Carlos Junior",
3   "email": "carlos.junior@gmail.com",
4   "password": "7654321"
5 }
```

- Response Status:** 201 Created
- Response Body (Pretty JSON):**

```
1 {
2   "message": "User registered successfully",
3   "user": {
4     "id": 11,
5     "user_id": "562e67a9-8c7e-4151-bcf2-67879288e258",
6     "name": "Carlos Junior",
7     "email": "carlos.junior@gmail.com",
8     "password": "$2b$10$cbANJDJa9qFkNl37bQm10pyjR6Ct2Z0tq4GG5fQXkxy.WXzAoGBe",
9     "preferences": null,
10    "loyalty_points": 0
11  }
12 }
```

Screenshot 2: Login Request

- Request URL:** https://backend-2-i0ej.onrender.com/login
- Method:** POST
- Body (raw JSON):**

```
1 {
2   "email": "carlos.junior@gmail.com",
3   "password": "7654321"
4 }
```

- Response Status:** 200 OK
- Response Body (Pretty JSON):**

```
1 {
2   "message": "Login successful.",
3   "user": {
4     "id": 11,
5     "user_id": "562e67a9-8c7e-4151-bcf2-67879288e258",
6     "name": "Carlos Junior",
7     "email": "carlos.junior@gmail.com"
8   }
8 }
```

The screenshot shows two separate Postman sessions.

Session 1: PUT update menu item

- Request:** PUT https://backend-2-i0ej.onrender.com/menu-item/093e3ff7-7439-4776-933a-b90e06974cb5
- Body (JSON):**

```

1 {
2   "name": "Rice and Okra Soup",
3   "description": "Enjoy your sunday afternoons like this",
4   "price": 18,
5   "image_url": "https://images.bolt.eu/store/2024-06-28/1a23c566-78ed-471c-857d-86cfb872d01d.jpeg",
6   "customizations": [
7     { "name": "Extra Fish", "price": 3 },
8     { "name": "Add Meat", "price": 4 }
9 ]

```

- Response:** 200 OK

Session 2: POST cart

- Request:** POST http://localhost:4000/cart
- Body (JSON):**

```

1 {
2   "user_id": "a4e4a5aa-39c6-4e1b-b4c5-ceb70f76e084",
3   "food_id": "d9ae330c-4705-46b4-8125-eed8017add02",
4   "quantity": 3,
5   "price": 18
6 }

```

- Response:** 201 Created

Frontend Design

- **Home Page:** Displays the menu items.
- **Cart Page:** Allows users to manage their cart.

Future Enhancements

1. Add real-time notifications for order status updates.
2. Implement a payment gateway for secure transactions.

3. Enable user reviews and ratings for menu items.

Conclusion

"Icy Foodies" is a robust food ordering platform that combines functionality, user engagement, and scalability. It is designed to provide a seamless experience for users and valuable insights for food businesses. By leveraging modern web technologies, "Icy Foodies" aims to set a new standard for online food services.